Chapter 2

Fundamentals of MATLAB Programming

Exercise 2.1 *In MATLAB environment, the following statements can be given*

tic, $\boldsymbol{A}$=rand(500); $\boldsymbol{B}$=inv($\boldsymbol{A}$); norm($\boldsymbol{A}$*$\boldsymbol{B}$-eye(500)), toc

Run the statements and observe results. If you are not sure with the commands, just use the on-line `help` *facilities to display information on the related functions. Then explain in detail the statement and the results.*

Solution What the statements actually do is to calculate and verify the inverse matrix $\boldsymbol{B}$ of a 500×500 randomly generated matrix $\boldsymbol{A}$, and measure the total time consumes. It can be found that the precision reaches 10^{-12}-level, and the time required is around one second.

Exercise 2.2 *Suppose that a polynomial can be expressed by $f(x) = x^5 + 3x^4 + 4x^3 + 2x^2 + 3x + 6$. If one wants to substitute x by $(s-1)/(s+1)$, the function $f(x)$ can be changed into a function of s. Use the Symbolic Math Toolbox to do the substitution and get the simplest result.*

Solution One should declare the two variables s and x as symbolic variables, then the `subs()` function should be used to do variable substitution. Finally, simplification of the results should be performed

```
>> syms s x, f=x^5+3*x^4+4*x^3+2*x^2+3*x+6;
   F=subs(f,x,(s-1)/(s+1)), F=simplify(F)
```

which leads to the result $F = \dfrac{3 + 23s + 54s^2 + 70s^3 + 19s^5 + 23s^4}{(s+1)^5}$.

Exercise 2.3 *Please simplify $\sin(k\pi + \pi/6)$, for any integer k.*

Solution In the new versions of MATLAB, integer symbolic variables are supported. The original problem can be solved with the following statements, with the result $(-1)^k/2$.

```
>> syms k, assume(k,'integer'); simplify(sin(k*pi+pi/6))
```

Exercise 2.4 *Input the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ into MATLAB workspace where*

$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \\ 2 & 3 & 4 & 1 \\ 3 & 2 & 4 & 1 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 1+\mathrm{j}4 & 2+\mathrm{j}3 & 3+\mathrm{j}2 & 4+\mathrm{j}1 \\ 4+\mathrm{j}1 & 3+\mathrm{j}2 & 2+\mathrm{j}3 & 1+\mathrm{j}4 \\ 2+\mathrm{j}3 & 3+\mathrm{j}2 & 4+\mathrm{j}1 & 1+\mathrm{j}4 \\ 3+\mathrm{j}2 & 2+\mathrm{j}3 & 4+\mathrm{j}1 & 1+\mathrm{j}4 \end{bmatrix}.$$

It is seen that $\boldsymbol{A}$ is a 4×4 matrix. If a command $\boldsymbol{A}(5,6) = 5$ is given, what will happen?

Solution The two matrices can be specified easily with the following statements

```
>> A=[1 2 3 4; 4 3 2 1; 2 3 4 1; 3 2 4 1]
   B=[1+4i 2+3i 3+2i 4+1i;  4+1i 3+2i 2+3i 1+4i;
      2+3i 3+2i 4+1i 1+4i; 3+2i 2+3i 4+1i 1+4i];
```

If further the command

```
>> A(5,6)=5
```

is used, and columns and rows in the statements are all greater than the current size of $\boldsymbol{A}$, zero terms are introduced to the extended part of $\boldsymbol{A}$, then the (5,6)th term is assigned to 5.

Exercise 2.5 *Command* $\boldsymbol{A}=$`rand(3,4,5,6,7,8,9,10,11)` *can be used to generate a multi-dimensional array. How many elements are there in the array? Please find the sum of all its elements.*

Solution The multi-dimensional array can be generated first, and the size of the array can be measured with the following statements, and it can be seen that there are a total of 19958400 elements in the array, and the sum under this run is about 9.9791×10^6. Since the quantities in the array are generated randomly, the sums under each run are different.

```
>> A=rand(3,4,5,6,7,8,9,10,11); prod(size(A)), sum(A(:))
```

Exercise 2.6 *Find the first 200 digits of the irrational numbers* $\sqrt{2}$, $\sqrt[6]{11}$, $\sin 1^\circ$, e^2, $\ln 21$.

Solution Since the quantities are irrational numbers, Symbolic Toolbox should be used to display all the first 200 digits. The commands to use are as follows, and the displays are omitted here.

```
>> e1=vpa(sqrt(sym(2)),200), e2=vpa(sym(11)^(1/6),200),
   e3=vpa(sin(sym(pi/180)),200), e4=vpa(exp(sym(2)),200),
   e5=vpa(log(sym(21)),200),
```

Exercise 2.7 *Please show the following identical equations*

$$(i)\ \mathrm{e}^{\mathrm{j}\pi}+1=0,\quad (ii)\ \frac{1-2\sin\alpha\cos\alpha}{\cos^2\alpha-\sin^2\alpha}=\frac{1-\tan\alpha}{1+\tan\alpha}.$$

Solution To show identities, the best way is to move everything to the left of the equation sign, and simplify the result. If the simplified quantities of the left-hand-side are zeros, the equations are proven. The two identical equations can be proven directly with the following statements

```
>> simplify(exp(sym(pi)*1i)+1)
   syms a;
   simplify((1-2*sin(a)*cos(a))/(cos(a)^2-sin(a)^2)...
        -(1-tan(a))/(1+tan(a)))
```

Exercise 2.8 *If* $f(x)=x^2-x-1$, *please find* $f(f(f(f(f(f(f(f(f(f(x))))))))))$, *and also express the result in a polynomial. What is the degree of the polynomial?*

Solution In recent versions of MATLAB, symbolic function definition of $f(x)$ is allowed. Therefore, the composite function can be constructed. The degree of the final polynomial is as high as 1024, and the expanded polynomial can be obtained, and omitted. It should be noted that when $f(x)$ is to be used, it should be declared with `syms` command first, otherwise it may not be recognized.

```
>> syms x f(x); f(x)=x^2-x-1; F=f(f(f(f(f(f(f(f(f(f(x)))))))))),
   expand(F), n=length(sym2poly(ans))-1
```

Exercise 2.9 *If the mathematical functions are known*

$$f(x)=\frac{x\sin x}{\sqrt{x^2+2}(x+5)},\ g(x)=\tan x,$$

please find out the functions $f(g(x))$ and $g(f(x))$.

Solution With the `subs()` function, variable substitutions can be made, to compute the two composite functions

```
>> syms x; f=x*sin(x)/sqrt(x^2+2)/(x+5);
   g=tan(x); F1=simplify(subs(f,x,g)), F2=simplify(subs(g,x,f))
```

and the results obtained are

$$F_1(x)=\frac{\sin(\tan x)\ \tan x}{(\tan x+5)\sqrt{\tan^2 x+2}},\ F_2(x)=\tan\left(\frac{x\sin x}{\sqrt{x^2+2}\,(x+5)}\right)$$

Alternatively, the following statements can be used in new the versions, and the same results can be obtained

```
>> syms x g(x) f(x); f(x)=x*sin(x)/sqrt(x^2+2)/(x+5);
   g(x)=tan(x); F1=simplify(f(g(x))), F2=simplify(g(f(x)))
```

Exercise 2.10 *Since double-precision scheme is quite limited in representing accurately large numbers, symbolic calculations are often used to find factorials of large numbers. Please use numerical and symbolic methods to calculate C_{50}^{10}, where, $\mathrm{C}_m^n=m!/[n!(m-n)!]$. Alternatively function* `nchoosek(sym(`*m*`),`*n*`)` *can be used.*

Solution The factorial $n!$ can be evaluated with `prod(1:`*n*`)`, and it is also possible to evaluated from `gamma(`*n*`+1)`. If n is very large, n should be converted to symbolic data type first. The following statements can be used

```
>> m=50; n=10; r1=gamma(m+1)/gamma(n+1)/gamma(m-n+1)
   m=sym(m); n=sym(n); r2=gamma(m+1)/gamma(n+1)/gamma(m-n+1)
```

and the numerical and analytical solutions can be obtained as $r_1=1.027227816999992\times 10^{10}$, $r_2=10272278170$. Alternatively, `nchoosek()` function can be used and the same result can be obtained.

```
>> r3=nchoosek(sym(m),n)
```

Exercise 2.11 *For a matrix $\boldsymbol{A}$, if one wants to extract all the even rows to form matrix $\boldsymbol{B}$, what command should be used? Suppose that matrix $\boldsymbol{A}$ is defined by $\boldsymbol{A}=$*`magic(8)`*, establish matrix $\boldsymbol{B}$ with suitable statements and see whether the results are correct or not.*

Solution Even row extraction of matrix $\boldsymbol{A}$ can easily be performed by

```
>> A=magic(8), B=A(2:2:end,:)
```

Exercise 2.12 *Please list all the positive integers such that it is a multiple of 11 and does not exceed 1000. Please also find all the integers which are multiples of 11 in the $[3000,5000]$ interval.*

Solution The first item is 11, and the others can be generated with colon expression, with an increment of 11. The expected integers can be obtained with

```
>> i1=11:11:1000
```

The integers in the interval $[3000,5000]$ can be extracted with the following statements

```
>> i2=11:11:5000; i2=i2(i2>=1000)
```

Exercise 2.13 *How many prime numbers are there in the interval* $[1, 1000000]$*? Please find the product of all the prime numbers in the interval. What is the number and how many digits are there? Measure the time elapsed in the evaluation.*

Solution There are 78498 prime numbers in the interval. The total number of the product has 433636 decimal digits, and it can be obtained in around 10 seconds.

```
>> length(primes(1000000)), tic, prod(primes(sym(1000000))); toc
   vpa(log10(ans))
```

Exercise 2.14 *It is known in Example 2.12 that* `gcd()` *and* `lcm()` *functions can be used to find the greatest common divisor and least common multiple of two entities only. Please write functions* `gcds()` *and* `lcms()` *such that arbitrary number of entities can be processed.*

Solution With the quantity `varargin`, the two functions can be written as follows. Both of the functions support arbitrary numbers of input arguments.

```
function k=gcds(varargin)
k=varargin{1}; for i=2:nargin, k=gcd(k,varargin{i}); end

function k=lcms(varargin)
k=varargin{1}; for i=2:nargin, k=lcm(k,varargin{i}); end
```

Having written the above functions, the least common multiple of 1, 2, 3, $\cdots$, 10, 11, 12 can be evaluated with the following statements, and the result is $K = 27720$.

```
>> K=lcms(1,2,3,4,5,6,7,8,9,10,11,12)
```

Exercise 2.15 *Implement the following piecewise function where* $\boldsymbol{x}$ *can be given by scalar, vectors, matrices or even other multi-dimensional arrays, the returned argument* $\boldsymbol{y}$ *should be the same size as that of* $\boldsymbol{x}$*. The parameters* h *and* D *are scalars.*

$$\boldsymbol{y} = f(\boldsymbol{x}) = \begin{cases} h, & \boldsymbol{x} > D \\ h/D\boldsymbol{x}, & |\,\boldsymbol{x}\,| \leqslant D \\ -h, & \boldsymbol{x} < -D. \end{cases}$$

Solution Two methods can be used, and the best one is with the use of the relationship expression in a clever way

```
>> y=h*(x>D) + h/D*x.*(abs(x)<=D) -h*(x<-D);
```

An alternative method is by the use of loops and condition structures

```
>> for i=1:length(x)
      if x(i)>D, y(i)=h;
      elseif abs(x(i))<=D, y(i)= h/D*x(i); else, y(i)=-h; end
   end
```

The structure of the latter statements are easy to understand, however the former is applicable not only for vector $\boldsymbol{x}$, but also to other data structures such as matrices or three-dimensional arrays.

Exercise 2.16 *A recursive formula is given by* $x_{n+1} = \dfrac{x_n}{2} + \dfrac{3}{2x_n}$*, with* $x_1 = 1$*. Please find* *a suitable number n, such that the terms after x_n approaches to a certain constant. The accuracy requirement is 10^{-14}. Please find also the constant.*

Solution The constant is also referred to as steady-state value, it means that $||x_{k+1}-x_k|| \leqslant \epsilon$, where ϵ is the pre-specified small number, for instant, 10^{-14}. It is not difficult to find the suitable $k = 6$, and the steady-state value obtained is $x_{k+1} = 1.7321$. It can be seen that the sequence approaches to the irrational number $\sqrt{3}$. The recursive formula is with quite high efficiency, since only 6 terms are needed to approach to the irrational $\sqrt{3}$.

```
>> k=1; x0=1;
   while (1), x1=x0/2+3/(2*x0);
      if abs(x1-x0)<1e-14, break; else, k=k+1; x0=x1; end
   end
```

Exercise 2.17 *Please calculate* $S = \prod_{n=1}^{\infty}\left(1+\dfrac{2}{n^2}\right)$, *under precision requirement of* $\epsilon = 10^{-12}$.

Solution It seems that the product can be obtained with loop structure, and the result thus obtained is 9.5648. Please pay attention to the condition $|1-p_1| \leqslant 10^{-10}$. It should not be written as $|p_1| \leqslant 10^{-12}$.

```
>> p=1;
   for n=1:10000, p1=(1+2/n^2); if abs(1-p1)<=1e-12, break; end
      p=p*p1;
   end
```

When checking the value of n, it is found that the $n = 10000$, which means that even all the terms are tested, the condition is not satisfied. To solve the accuracy problem, more terms must be tried, for instance, set the terms to 10000000000, or a larger number. While this method is rather time consuming. The actual n is $n = 1414230$, and $p = 9.5667$.

```
>> p=1;
   for n=1:100000000000000,
      p1=(1+2/n^2); if abs(1-p1)<=1e-12, break; end
      p=p*p1;
   end
```

Exercise 2.18 *It is known that*

$$\arctan(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \cdots.$$

Let $x = 1$, the following formula can be derived

$$\pi \approx 4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \cdots\right).$$

Please calculate the approximate value of π, with precision requirement 10^{-6}.

Solution With the accumulation approach below, the approximate value of π can be found, and it is quite time consuming, in this example, 4 seconds are needed, since 500001 terms are needed to meet the requirement. If the error tolerance is set to an even smaller number, much more time is needed. The readers may try to change the `1e-6` term to `1e-8` and see what happens - 4 minutes on my computer.

```
>> s=0; s0=1; tic
```

```
for i=1:100000000,
   s1=1/(2*i-1)*s0; s0=-s0;
   if abs(s1)<1e-6, break; end, s=s+s1;
end
toc, S=4*s, i
```

Exercise 2.19 *Generate a* 100×100 *magic matrix, and find out the entities greater than 100, and substitute the entities by 0.*

Solution The magic matrix can be generated, and all the terms greater than 100 can be found, and set by force with the following statements

```
>> A=magic(100); i=find(A>100); A(i)=0;
```

Exercise 2.20 *Evaluate using numerical method the sum*

$$S = 1 + 2 + 4 + \cdots + 2^{62} + 2^{63} = \sum_{i=0}^{63} 2^i,$$

the use of vectorized form is suggested. Check whether accurate solutions can be found and why. Find the accurate sum using the symbolic computation methods. What happened in numerical and analytical solutions, if the number of terms is increased to 640.

Solution The following statements can be used to evaluate numerically the sum, however due to the limitations of the 64-bit `double` data type, the result $s_1 = 1.844674407370955 \times 10^{19}$ is not accurate.

```
>> s1=sum(2.^[0:63])
```

To solve the problem accurately, the symbolic data type should be used instead. The new statement should be

```
>> s2=sum(sym(2).^[0:63])
```

with $s_2 = 18446744073709551615$. One may even replace the term 63 by 1000 to evaluate accurately $\sum_{i=0}^{1000} 2^i$, which is not possible by numerical data types. The accurate result is
21430172143725346418968500981200036211228096234110672148875007767407021022498722
44986396757631391716255189345835106293650374290571384628087196915514939714960786
91355496484619708421492101247422837559083643060929499671638825347975351183310878
92154125829142392955373084335320859663305248773674411336138751.

Exercise 2.21 *Please use two algorithms to solve the equation* $f(x) = x^2 \sin(0.1x+2) - 3 = 0$.

(i) **Bisection method**. *If in an interval* (a, b), $f(a)f(b) < 0$, *there will be at least one solution. Take the middle point* $x_1 = (b - a)/2$, *and based on the relationship of* $f(x_1)$ *and* $f(a)$, $f(b)$, *determine in which half interval there exists solutions. Middle point in the new half interval can then be taken. Repeat the process until the size of the interval is smaller than the pre-specified error tolerance* ϵ. *Find the solution with bisection method in interval* $(-4, 0)$, *with* $\epsilon = 10^{-10}$.

(ii) **Newton–Raphson method**. *Select an initial guess of* x_n, *the next approximation can be obtained with* $x_{n+1} = x_n - f(x_n)/f'(x_n)$. *If the two points are close enough, i.e.,* $|x_{n+1} - x_n| < \epsilon$, *where* ϵ *is the error tolerance. Find the solution with* $x_0 = -4$, *and* $\epsilon = 10^{-12}$.

Solution No matter which algorithm is adopted, the equation should be described as an anonymous function in MATLAB as

```
>> f=@(x)x.^2.*sin(0.1*x)-3;
```

(i) **Bisection algorithm**. Bisection algorithm can be used to solve the equation, and the function can be written as follows. A central point x in the interval can be obtained, and the value of the function can be compared with $f(a)$ and $f(b)$. If the signs are different, the values of a or b are updated, until finally a very small interval can be found, and the central point can be regarded as the root.

```
function x=p_math_2_18a(f,a,b,err)
if f(a)*f(b)>0,
  error('bisectional method failed since f(a)*f(b)>0.'); end
while abs(b-a)>err, x=(a+b)/2;
   if f(x)*f(a)<0, b=x; else, a=x; end
end
```

The root of the equation can be obtained as $x=3.1242$, and the value of the function at this point is $f(x)=-2.2\times 10^{-11}$.

```
>> x=p_math_2_18a(f,0,4,1e-10), f(x)
```

(ii) **Gradient algorithm**: Since the derivative $f'(x)$ is needed, and it can be obtained as $f'(x) = 2x\sin(x/10) + (x^2\cos(x/10))/10$, the function can be expressed with anonymous function

```
>> syms x; diff(x^2*sin(0.1*x)-3),
   f1=@(x)2*x.*sin(x/10)+(x.^2.*cos(x/10))/10;
```

Alternatively, gradient algorithm can be used to solve the equation

```
function x=p_math_2_18b(f,f1,x0,err)
while (1), x=x0-f(x0)/f1(x0);
   if abs(f(x))>err, x0=x; else, break; end
end
```

The following statements can be used to find the result $x = 3.1242$, with $f(x) = 1.83\times 10^{-12}$.

```
>> x=p_math_2_18b(f,f1,-4,1e-10), f(x)
```

Exercise 2.22 *Write an M-function* `mat_add()` *with the* $\boldsymbol{A}=$`mat_add(`$\boldsymbol{A}_1,\boldsymbol{A}_2,\boldsymbol{A}_3,\cdots$`)` *syntax. It is required that arbitrary number of input arguments* $\boldsymbol{A}_i$ *are allowed.*

Solution With the use of `varargin`, the function below can be designed.

```
function A=mat_add(varargin)
A=0; for i=1:length(varargin), A=A+varargin{i}; end
```

The `try-catch` structure can further be used to solve the above problem.

```
function A=mat_add(varargin)
try
   A=0; for i=1:length(varargin), A=A+varargin{i}; end
catch, error(lasterr); end
```

Exercise 2.23 *A MATLAB function can be written whose syntax is*

$$\boldsymbol{v} = [h_1, h_2, h_m, h_{m+1}, \cdots, h_{2m-1}] \quad \textit{and} \quad \boldsymbol{H} = \texttt{myhankel}(\boldsymbol{v})$$

where the vector $\boldsymbol{v}$ *is defined, and out of it, the output argument should be an* $m \times m$ *Hankel matrix.*

Solution Many methods can be used to solve the above problem:

(i) The most straightforward method is the use of double loop structure to implement $H_{i,j} = h_{i+j-1}$ such that

```
function H=myhankel(v)
m=(length(v)+1)/2;
for i=1:m, for j=1:m, H(i,j)=v(i+j-1); end, end
```

(ii) For a certain column (or row), $\boldsymbol{a}_i = [h_i, h_{i+1}, \cdots, h_{i+m-1}]$. Thus single loop structure can be used to generate the Hankel matrix

```
function H=myhankel(v)
m=(length(v)+1)/2; for i=1:m, H(i,:)=v(i:i+m-1); end
```

(iii) Based on the existing `hankel()` function, one can write

```
function H=myhankel(v)
m=(length(v)+1)/2; H=hankel(v(1:m),v(m:end));
```

Exercise 2.24 *From matrix theory, it is known that if a matrix* $\boldsymbol{M}$ *is expressed as* $\boldsymbol{M} = \boldsymbol{A} + \boldsymbol{B}\boldsymbol{C}\boldsymbol{B}^{\mathrm{T}}$, *where* $\boldsymbol{A}$, $\boldsymbol{B}$ *and* $\boldsymbol{C}$ *are the matrices of relevant sizes, the inverse of* $\boldsymbol{M}$ *can be calculated by the following algorithm*

$$\boldsymbol{M}^{-1} = \left(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{C}\boldsymbol{B}^{\mathrm{T}}\right)^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{B}\left(\boldsymbol{C}^{-1} + \boldsymbol{B}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{A}^{-1}$$

The matrix inversion can be carried out using the formula easily. Suppose that there is a 5×5 *matrix* $\boldsymbol{M}$, *from which the three other matrices can be found.*

$$\boldsymbol{M} = \begin{bmatrix} -1 & -1 & -1 & 1 & 0 \\ -2 & 0 & 0 & -1 & 0 \\ -6 & -4 & -1 & -1 & -2 \\ -1 & -1 & 0 & 2 & 0 \\ -4 & -3 & -3 & -1 & 3 \end{bmatrix}, \quad \boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 0 & 1 \\ 1 & 1 & 1 & 2 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 1 & -1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & -1 & 0 & 1 \end{bmatrix}.$$

Write the statement to evaluate the inverse matrix. Check the accuracy of the inversion. Compare the accuracy of the inversion method and the direct inversion method with `inv()` *function.*

Solution Based on the partition formula, the following function can be written

```
function Minv=part_inv(A,B,C)
Minv=inv(A)-inv(A)*B*inv(inv(C)+B'*inv(A)*B)*B'*inv(A);
```

For the given matrices, the following two methods can be used

```
>> M=[-1,-1,-1,1,0; -2,0,0,-1,0; -6,-4,-1,-1,-2;
      -1,-1,0,2,0;-4,-3,-3,-1,3];
   A=[1,0,0,0,0; 0,3,0,0,0; 0,0,4,0,0; 0,0,0,2,0; 0,0,0,0,4];
   B=[0,1,1,1,1; 0,2,1,0,1; 1,1,1,2,1; 0,1,0,0,1; 1,1,1,1,1];
   C=[1,-1,1,-1,-1; 1,-1,0,0,-1; 0,0,0,0,1; 1,0,-1,-1,0; 0,1,-1,0,1];
   M1=inv(M), % method 1, direct numerical solution
   M2=part_inv(A,B,C)   % method 2, with partition formula
   Ms=inv(sym(M)); e1=norm(double(Ms)-M1), e2=norm(double(Ms)-M2)
```

The appearance of $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are the same, however, the precision might be different, since $e_1 = 1.5232{\times}10^{-16}, e_2 = 1.5271{\times}10^{-15}$. It can be concluded that normally when direct functions exist, it should be used, rather than using any other indirect methods, to avoid accumulative errors.

Exercise 2.25 *Please generate the first 300 terms of the extended Fibonacci sequence* $T(n) = T(n-1) + T(n-2) + T(n-3), n = 4, 5, \cdots$, *with* $T(1) = T(2) = T(3) = 1$.

Solution The best way to solve the problem is to use the loop structure, and it is found that the last term is 587878818669102731346904754726368926963546999305879075592534791178 5909345498275.

```
>> T=sym([1 1 1]);
   for i=4:300, T(i)=T(i-1)+T(i-2)+T(i-3); end
```

Please note that recursive structure is not recommended, and in fact, it cannot be used in solving such a problem.

Exercise 2.26 *Consider the following iterative model*

$$\begin{cases} x_{k+1} = 1 + y_k - 1.4x_k^2 \\ y_{k+1} = 0.3x_k \end{cases}$$

with initial conditions $x_0 = 0$, $y_0 = 0$. *Write an M-function to evaluate the sequence* x_i, y_i. *30000 points can be obtained by the function to construct the* $\boldsymbol{x}$ *and* $\boldsymbol{y}$ *vectors. The points can be expressed by a dot, rather than lines. In this case, the so-called* Hénon attractor *can be drawn.*

Solution Loop structure can be used to implement the recursive formula, and the Hénon attractor can be drawn as shown in Figure 2.1. Note that, the option `'.'` should be used to indicate the sequence.

```
>> n=30000; x=zeros(1,n); y=x;
   for i=1:n-1, x(i+1)=1+y(i)-1.4*x(i)^2; y(i+1)=0.3*x(i); end
   plot(x,y,'.')
```

Exercise 2.27 *The well-known* Mittag–Leffler function *is defined as*

$$f_\alpha(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + 1)},$$

where $\Gamma(x)$ *is* Γ*-function which can be evaluated with* `gamma(`x`)`. *Write an M-function with syntax* $\boldsymbol{f}$ = `mymittag(`α`,`$\boldsymbol{z}$`,`ϵ`)`, *where* ϵ *is the error tolerance, with default value of* $\epsilon = 10^{-6}$. *Argument* $\boldsymbol{z}$ *is a numeric vector. Draw the curves for Mittag–Leffler functions with* $\alpha = 1$ *and* $\alpha = 0.5$.

Solution The `while` loop structure can be used for the problem. If the norm of the

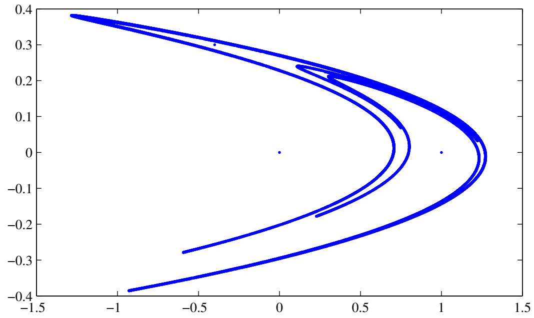


FIGURE 2.1: Hénon attractor

increment is smaller than the pre-specified small value ϵ, the loop can be terminated, and the result can be returned.

```
function f=mymittag(a,z,err)
f=0; k=0; if nargin==2, err=1e-6; end
while (1)
   df=z.^k/gamma(a*k+1);
   if norm(df)>err, f=f+df; k=k+1; else, break; end
end
```

With the above function, the two Mittag–Leffler function curves can be obtained as shown in Figure 2.2. The curve for $\alpha = 1$ is equivalent to exponential function e^z. Better Mittag–Leffler function implementation will be given in Chapter 8 of the book.

```
>> z=0:0.01:1; y1=mymittag(1,z,1e-6);
   y2=mymittag(0.5,z,1e-6); plot(z,y1,'-',z,y2,'--')
```

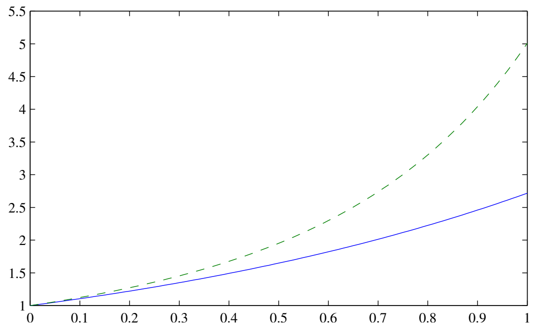


FIGURE 2.2: Two Mittag–Leffler function curves

Exercise 2.28 Chebyshev polynomials *are mathematically defined as*

$$T_1(x) = 1,\ T_2(x) = x,\ T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x),\ n = 3, 4, 5, \cdots .$$

Please write a recursive function to generate a Chebyshev polynomial, and compute $T_{10}(x)$*. Please write a more efficient function as well to generate Chebyshev polynomials, and find* $T_{30}(x)$*.*

Solution Two MATLAB functions can be written to generate Chebyshev polynomials, while in the first function, recursive structure is used, and in the second one, loop structure is used. It is recommended to use the second function, since the recursive structure is time consuming and cannot be used for large N.

```
function T=chebyshev_poly(n,x)
if n==0, T=1; elseif n==1, T=x;
else, T=2*x*chebyshev_poly(n-1,x)-chebyshev_poly(n-2,x); end

function T=chebyshev_poly1(n,x)
T0=1; T1=x; for k=3:n+1, T=collect(2*x*T1-T0); T0=T1; T1=T; end
```

The Chebyshev polynomial $T_{30}(x)$ can be obtained with

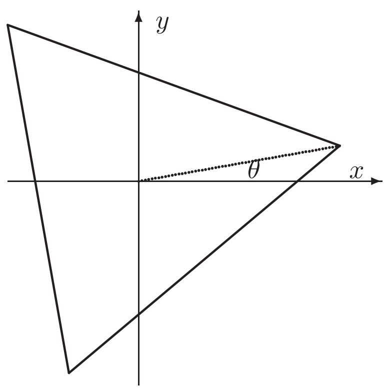
```
>> syms x; T=chebyshev_poly1(30,x), collect(T)
```

Exercise 2.29 *A regular triangle can be drawn by MATLAB statements easily. Use the loop structure to design an M-function that, in the same coordinates, a sequence of regular triangles can be drawn, each by rotating a small angle, for instance, 5°, from the previous one.*

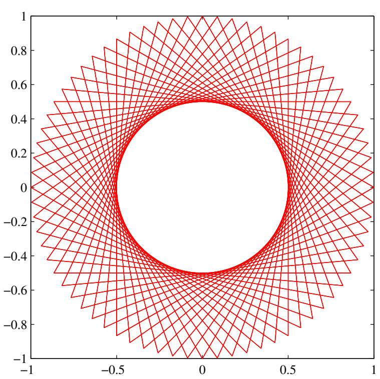
Solution To rotate counter-clockwise an equilateral triangle by the angle θ, the new triangle can be illustrated as shown in Figure 2.3 (a). The critical points of the new triangle are respectively $(\cos\theta, \sin\theta)$, $(\cos(\theta+120°), \sin(\theta+120°))$ and $(\cos(\theta+240°), \sin(\theta+240°))$, then back to point $(\cos\theta, \sin\theta)$. The triangle can be drawn easily. Increment the angle θ continuously and draw in a loop a set of triangles, the resulted graphical display is shown in Figure 2.3 (b), with the command

```
>> draw_triangles(5,'r') % 5 and 'r' for increment angle and color
```

and the M-function `draw_triangles()` is listed below



(a) sketch of the rotation



(b) final results

FIGURE 2.3: Graphical display of a set of triangles

```
function draw_triangles(delta,col)
t=[0,120,240,0]*pi/180; xxx=[]; yyy=[];
for i=0:delta:360
   tt=i*pi/180; xxx=[xxx; cos(tt+t)]; yyy=[yyy; sin(tt+t)];
end
plot(xxx',yyy',col), axis('square')
```

Selecting the increment to other values such as $\Delta\theta = 2, 1, 0.1$, one may further observe the results.

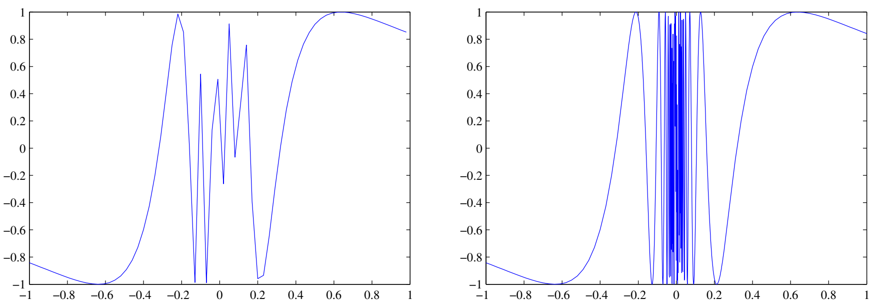
Exercise 2.30 *Select suitable step-sizes and draw the function curve for* $\sin(1/t)$, $t \in (-1, 1)$.

Solution In ordinary graphics mode, when a step-size of 0.03 is used, the curve of the function is shown in Figure 2.4 (a). However the curve is harsh.

```
>> t=-1:0.03:1; y=sin(1./t); plot(t,y)
```

If variable step-size is used, the curves are shown in Figure 2.4 (b).

```
>> t=[-1:0.03: -0.25, -0.248:0.001:0.248, 0.25:.03:1];
   y=sin(1./t); plot(t,y)
```



(a) fixed step-size (b) variable step-size

FIGURE 2.4: The curves of $\sin(1/t)$ under different step-sizes

It can be concluded from the example that, the curves obtained should be verified, before it can be put into practical used.

Exercise 2.31 *For suitably assigned ranges of* θ, *draw polar plots for the following functions.*

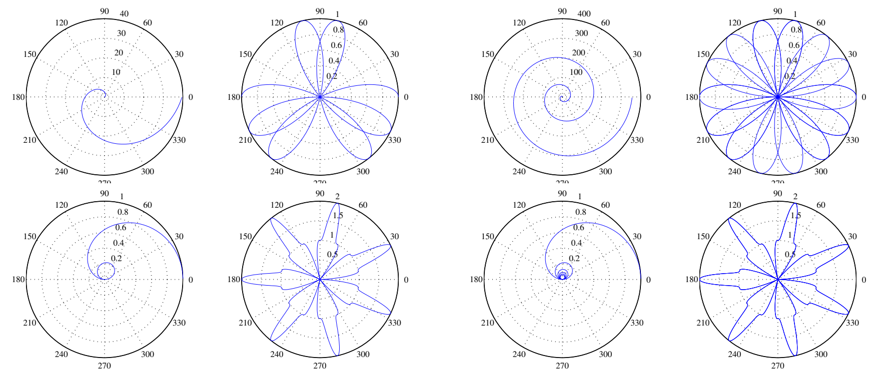
(i) $\rho = 1.0013\theta^2$, *(ii)* $\rho = \cos(7\theta/2)$, *(iii)* $\rho = \sin(\theta)/\theta$, *(iv)* $\rho = 1 - \cos^3(7\theta)$.

Solution It seems that drawing polar plot is an easy task. However one should be very careful in verifying the results by choosing different θ intervals. Also dot operation of vectors must be used. Comparing the plots for different θ ranges shown in Figures 2.5 (a) and (b). It can be seen that when the range of θ is small, some of the polar plots may not be complete.

```
>> t=0:0.01:2*pi; subplot(221), polar(t,1.0013*t.^2),  % (i)
   subplot(222), polar(t,cos(7*t/2)) % (ii)
   subplot(223), polar(t,sin(t)./t)                    % (iii)
   subplot(224), polar(t,1-(cos(7*t)).^3)              % (iv)
   figure; t=0:0.01:6*pi; % repeat the previous commands, get figure (b)
```

Exercise 2.32 *Please draw the curves of* $x \sin x + y \sin y = 0$ *for* $-50 \leqslant x, y \leqslant 50$.

Solution This is an implicit function, and the plot of the function can be drawn easily with the following statements, as shown in Figure 2.6.



(a) $\theta \in (0, 2\pi)$ (b) $\theta \in (0, 6\pi)$

FIGURE 2.5: Polar plots for different θ ranges

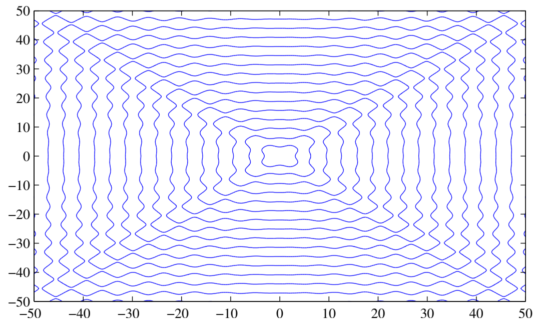


FIGURE 2.6: Curve of the implicit function

```
>> ezplot('x*sin(x)+y*sin(y)',[-50,50])
```

Exercise 2.33 *Find the solutions to the following simultaneous equations using graphical methods and verify the solutions.*

$$\text{(i)} \begin{cases} x^2+y^2=3xy^2 \\ x^3-x^2=y^2-y, \end{cases} \qquad \text{(ii)} \begin{cases} \mathrm{e}^{-(x+y)^2+\pi/2}\sin(5x+2y)=0 \\ (x^2-y^2+xy)\mathrm{e}^{-x^2-y^2-xy}=0. \end{cases}$$

Solution (i) The two equations can all be expressed by implicit function drawing command `ezplot()`, and the intersections are the solutions of the simultaneous equations, as shown in Figure 2.7 (a). One may zoom the plots and find more accurate values of the intersections.

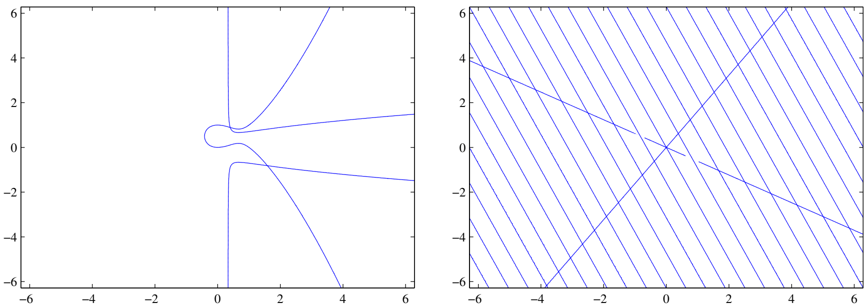
```
>> ezplot('x^2+y^2-3*x*y^2'); hold on, ezplot('x^3-x^2=y^2-y')
```

(ii) The equations can be drawn with the following statements, and the curves of the equations are as shown in Figure 2.7 (b).

```
>> ezplot('exp(-(x+y)^2+pi/2)*sin(5*x+2*y)'), hold on
   ezplot('(x^2-y^2+x*y)*exp(-x^2-y^2-x*y)')
```

Exercise 2.34 *Please save a* 100×100 *magic matrix into an Excel file.*

Solution The matrix can be generated and saved in the Excel file, with the following statements.



(a) equations in (i) (b) equations in (ii)

FIGURE 2.7: Graphical interpretations of the solutions

```
>> A=magic(100); xlswrite('myfile.xls',A)
```

Exercise 2.35 *Assume that the power series expansion of a function is*

$$f(x) = \lim_{N\to\infty} \sum_{n=1}^{N} (-1)^n \frac{x^{2n}}{(2n)!}.$$

If N is large enough, power series $f(x)$ converges to a certain function $\hat{f}(x)$. Please write a MATLAB program that plots the function $\hat{f}(x)$ in the interval $x \in (0, \pi)$. Observe and verify what function $\hat{f}(x)$ is.

Solution From the general term $g_n(x) = x^{2n}/(2n)!$, it is easily found that the accumulative relationship is $g_{n+1}(x)/g_n(x) = -x^2/(2n(2n-1))$, therefore selecting an error tolerance $\epsilon = 10^{-10}$, with loop structure, the function can be evaluated with

```
>> x=linspace(0,pi,50); y=zeros(size(x));
   e=1e-10; dg=ones(size(x)); n=0;
   while norm(dg)>e
      n=n+1; dg=-dg.*x.^2/2/n/(2*n-1); y=y+dg;
   end
   plot(x,y)
```

The plot obtained is as shown in Figure 2.8 and it seems that the actual function is $y(x) = \cos x - 1$.

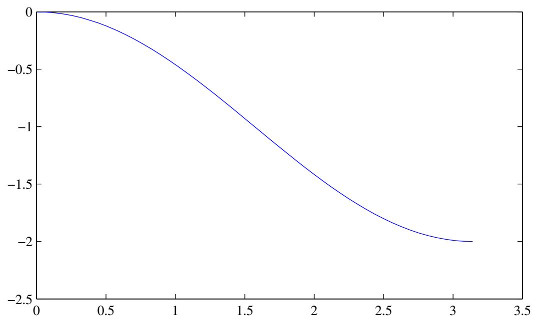


FIGURE 2.8: Obtained curve of the function

Exercise 2.36 *Draw the 3D surface plots for the functions xy and $\sin xy$ respectively. Also draw the contours of the functions. View the 3D surface plot from different angles, especially with orthographic views.*

Solution The following commands can be used to draw the 3D surface and contour lines of the functions. The function `view()` can be used to change view points and also one can rotate the 3D surfaces manually. The surfaces and contours of the two functions can be obtained as shown in Figure 2.9. It is interesting to note that when x and y are small — not necessarily approach zero, the values of the functions xy and $\sin xy$ are similar.

```
>> [x,y]=meshgrid(-1:.1:1); z1=x.*y; z2=sin(z1);
   subplot(211), surf(x,y,z1), subplot(212), contour(x,y,z1,30)
   figure;
   subplot(211), surf(x,y,z2), subplot(212), contour(x,y,z2,20)
```

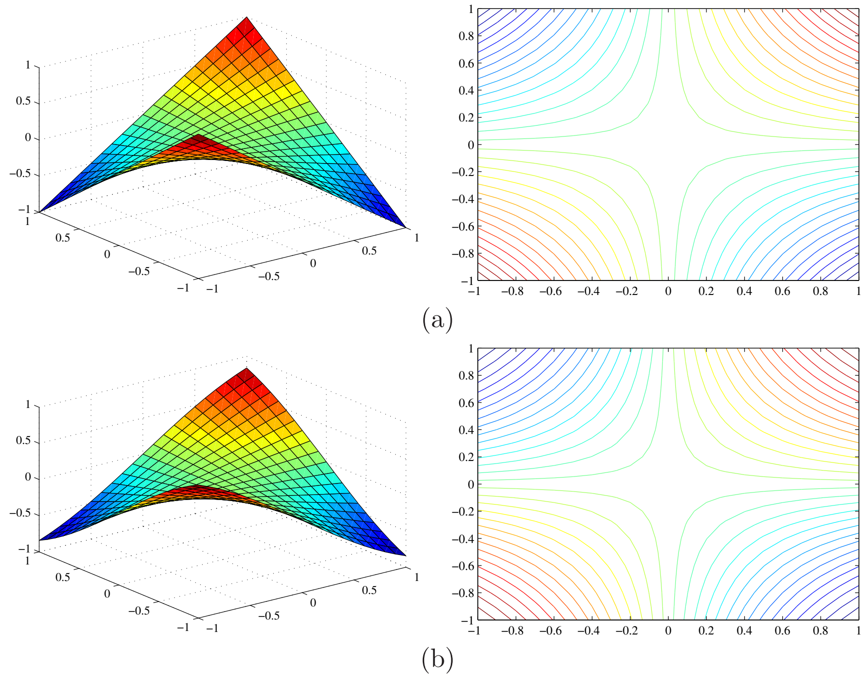


FIGURE 2.9: Surfaces and contours

Exercise 2.37 *Please draw 2D and 3D Lissajous figures under the parametric equations $x = \sin t, y = \sin at$, and $z = \sin bt$, for different parameters a and b, where, please try the following rational and irrational parameters and see what may happen.*

(i) $a = 1/2$, $b = 1/3$, *(ii)* $a = \sqrt[8]{2}$, $b = \sqrt{3}$.

Solution (i) For rational a, no matter how large the range of t is, neat phase plot shown in Figure 2.10 can be obtained.

```
>> t=0:0.01:10000; a=1/2; b=1/3; plot(sin(a*t),sin(t*b))
```

(ii) If a and/or b are irrational numbers, when the range of t is selected very large, the phase plot will fill the whole $(-1, 1)$ square in the x-y plane. To have a better understanding of the process, it is strongly recommended to use `comet()` function to replace `plot()`.

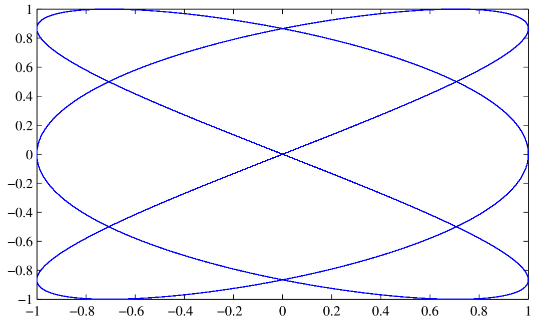


FIGURE 2.10: Phase plot

```
>> a=2^(1/8); b=sqrt(3); plot(sin(a*t),sin(t*b))
```

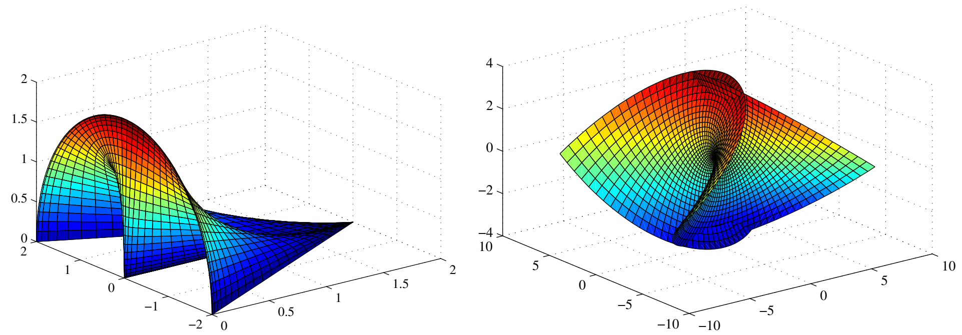
Exercise 2.38 *For the parametric equations* [1]*, please draw the surfaces*

(i) $x = 2\sin^2 u\cos^2 v, y = 2\sin u\sin^2 v, z = 2\cos u\sin^2 v, -\pi/2 \leqslant u, v \leqslant \pi/2,$

(ii) $x = u - \dfrac{u^3}{3} + uv^2,\ \ y = v - \dfrac{v^3}{3} + vu^2,\ \ z = u^2 - v^2,\ \ -2 \leqslant u, v \leqslant 2.$

Solution The parametric equations can be drawn directly with the following statements, as shown in Figure 2.11 (a).

```
>> syms u v; x=2*sin(u)^2*cos(v)^2; y=2*sin(u)*sin(v)^2;
   z=2*cos(u)*sin(v)^2; ezsurf(x,y,z,[-pi/2,pi/2,-pi/2,pi/2])
```



(a) surface in (i) (b) surface in (ii)

FIGURE 2.11: Graphical interpretations of the solutions

(ii) Similarly, the parametric equations can be drawn easily with the following statements, as shown in Figure 2.11 (b).

```
>> x=u-u^3/3+u*v^2; y=v-v^3/3+v*u^2; z=u^2-v^2;
   ezsurf(x,y,z,[-2,2,-2,2])
```

Exercise 2.39 *A vertical cylinder can be described by parametric equation* $x = r\sin u, y = r\cos u, z = v$*, along* z *axis, with* r *the radius. If* x *and* z *are swopped, the cylinder can be* *represented along x axis. Please draw several cylinders with different radii and directions together in the same coordinate.*

Solution The cylinders can be drawn with the following statements, as shown in Figure 2.12.

```
>> syms u v; r=1; x=r*sin(u); y=r*cos(u); z=v;
   ezsurf(x,y,z,[-2,2]), hold on
   x=0.5*sin(u); y=v; z=0.5*cos(u); ezsurf(x,y,z,[-4,4])
```

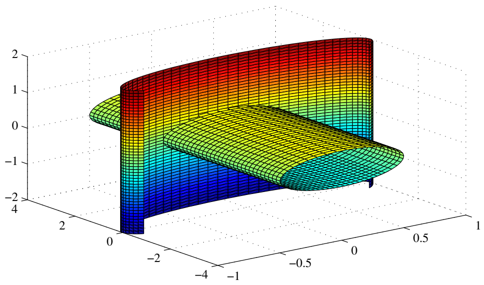


FIGURE 2.12: The intersection of the two cylinders

Exercise 2.40 *Please draw a cone, whose top is at (0,0,2), and bottom at the plane $z = 0$, with a radius of 1.*

Solution The edge of the cone can be calculated first, and the cone can be drawn with the following statements, as shown in Figure 2.13.

```
>> z0=0:0.1:1; r=1-z0; [x,y,z]=cylinder(r,100); surf(x,y,2*z)
```

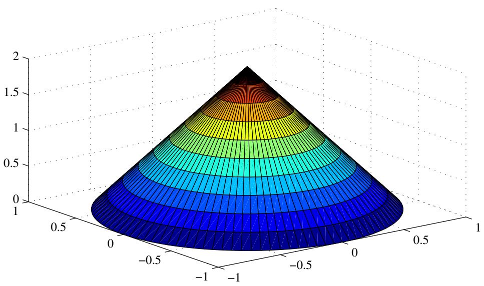


FIGURE 2.13: The surface of the cone

Exercise 2.41 *In graphics command, there is a trick in hiding certain parts of the plot. If the function values are assigned to* `NaN`*s, the point on the curve or the surface will not be shown. Draw first the surface plot of the function $z = \sin xy$. Then cut off the region that satisfies $x^2 + y^2 \leqslant 0.5^2$.*

Solution The mesh grid data of a rectangular region can be generated first and the function values can be calculated. Then find all the points in the region satisfying $x^2+y^2 \leqslant 0.5^2$, and set the values to `NaN`'s. The 3D surface of the given function excluding the region, shown in Figure 2.14.

```
>> [x,y]=meshgrid(-1:.1:1); z=sin(x.*y);
   ii=find(x.^2+y.^2<=0.5^2); z(ii)=NaN; surf(x,y,z)
```

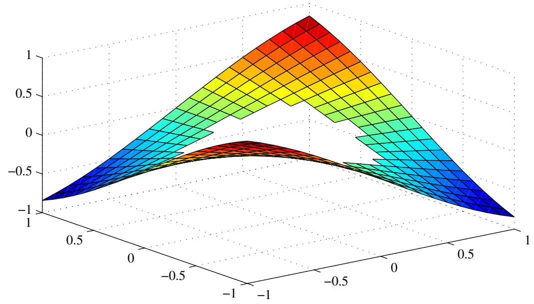


FIGURE 2.14: 3D surface with a region cut off

Exercise 2.42 *Please draw the 3D surface of* $f(x,y)=\dfrac{\sin\sqrt{x^2+y^2}}{\sqrt{x^2+y^2}}$ *for* $-8\leqslant x,y\leqslant 8$.

Solution For simplicity, it is better to use the function `ezsurf()` function to draw the surface plot, as shown in Figure 2.15.

```
>> ezsurf('sin(sqrt(x^2+y^2))/sqrt(x^2+y^2)',[-8,8])
```

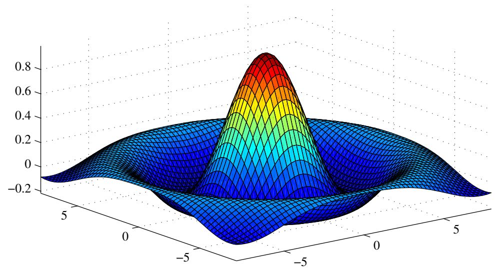


FIGURE 2.15: The 3D surface plot

Exercise 2.43 *Lambert W function is a commonly used special function, its mathematical form is* $W(z)\mathrm{e}^{W(z)}=z$. *Please draw the curve of the function.*

Solution Denote $y=W(z)$, the Lambert W function can be written as an equation of y, where $y\mathrm{e}^y=x$, where $x=z$ is the independent variable. In this case, the curve of Lambet W function can be drawn with the following command, as shown in Figure 2.16.

```
>> ezplot('y*exp(y)=x')
```

Exercise 2.44 *Draw the surface plot and contour plots for the following functions. Draw also with the functions* `surfc()`, `surfl()`, *and* `waterfall()`, *and observe the results.*

$$\text{(i)}\quad z=xy,\quad \text{(ii)}\quad z=\sin x^2y^3,\quad \text{(iii)}\quad z=\frac{(x-1)^2y^2}{(x-1)^2+y^2},\quad \text{(iv)}\quad z=-xy\,\mathrm{e}^{-2(x^2+y^2)}.$$

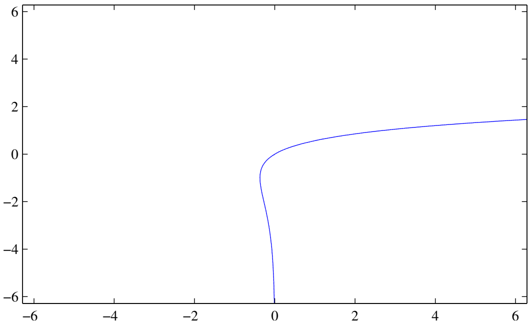


FIGURE 2.16: The curve of Lambert W function

Solution Take (ii) for example. Mesh grid can be generated first in the interested area, for instance, $(-1, 1)$, and the value of the function can be computed, and various 3D plots can be obtained as shown in Figure 2.17.

```
>> [x y]=meshgrid(-1:0.1:1); z=sin(x.^2.*y.^3);
   subplot(221), surf(x,y,z), subplot(222), surfl(x,y,z),
   subplot(223), surfc(x,y,z), subplot(224), waterfall(x,y,z),
```

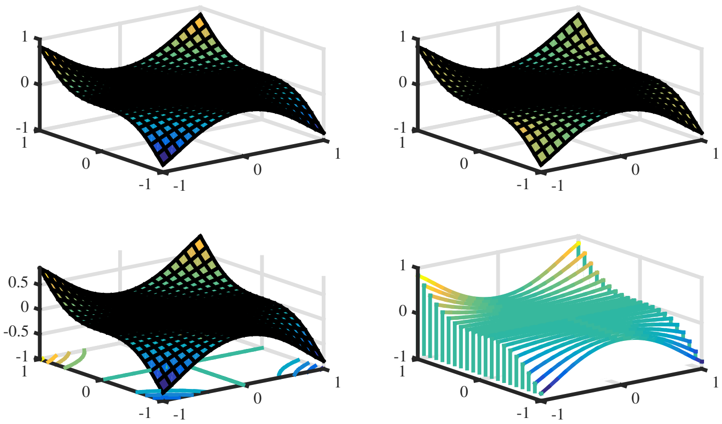


FIGURE 2.17: Different representation of the 3D surface

Exercise 2.45 *Please draw the surface of the three-dimensional implicit function*

$$(x^2 + xy + xz)\mathrm{e}^{-z} + z^2yx + \sin(x + y + z^2) = 0,$$

Solution The third-party `ezimplot3()` function can be used to draw the surface of the implicit function, as shown in Figure 2.18.

```
>> ezimplot3('(x^2+x*y+x*z)*exp(-z)+z^2*y*x+sin(x+y+z^2)')
```

Exercise 2.46 *Please draw the two following two surfaces and observe the intersections*

$$x^2 + y^2 + z^2 = 64,\ \ y + z = 0.$$

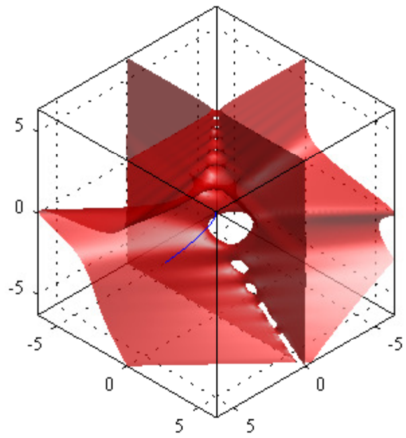


FIGURE 2.18: The 3D implicit function curve

Solution The two implicit functions can be drawn in the same coordinate, with the following statements, as shown in Figure 2.19.

```
>> ezimplot3('x^2+y^2+z^2-64'), ezimplot3('y+z')
```

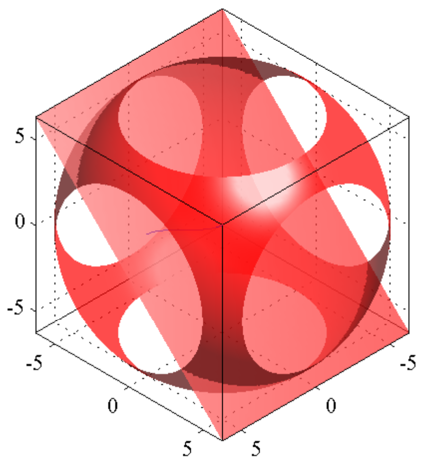


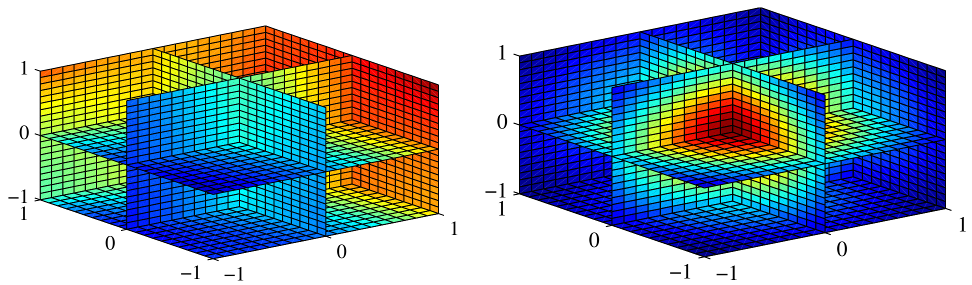
FIGURE 2.19: The 3D implicit function curve

Exercise 2.47 *Please draw the sliced volume visualization for the following functions*

(i) $V(x,y,z)=\sqrt{\mathrm{e}^{x}+\mathrm{e}^{(x+y)-xy}+\mathrm{e}^{(x+y+z)/3-xyz}}$, *(ii)* $V(x,y,z)=\mathrm{e}^{-x^2-y^2-z^2}$.

Solution The 3D grid data can be generated first, and the values of the function can be evaluated with dot operations. Volume visualization with slices from the data can be performed easily with the following statements, as shown in Figure 2.20.

```
>> [x y z]=meshgrid(-1:0.1:1);
   V=sqrt(exp(x)+exp((x+y)-x.*y)+exp((x+y+z)/3-x.*y.*z));
   slice(x,y,z,V,[0 1],[0 1],[-1,0])
   figure; V=exp(-x.^2-y.^2-z.^2); slice(x,y,z,V,[0 1],[0 1],[-1,0])
```



(a) function in (i) (b) function in (ii)

FIGURE 2.20: Volume visualization with slices

Alternatively, with the `vol_visual4d()` interface designed in the book, the volume visualization with standard slices for function in (ii) can be obtained directly, as shown in Figure 2.21.

```
>> vol_visual4d(x,y,z,V)
```

The slices can be adjusted easily with the controls in the interface.

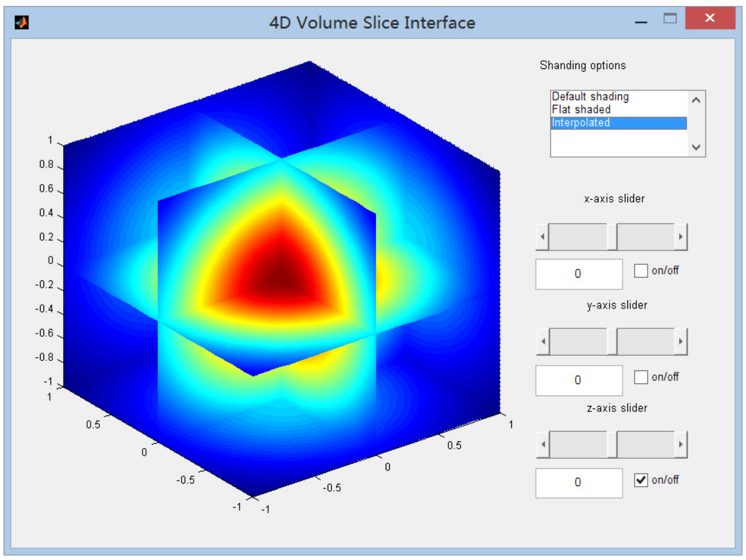


FIGURE 2.21: Slices in the example